AMENDMENTS TO THE CLAIMS

Claims 1, 2, 4, 13, 14, and 16 are pending in the subject application. It is requested that each of claims 1, 2, 13, and 14 be amended as set forth herein. Upon entry of the amendments, claims 1, 2, 4, 13, 14, and 16 will remain pending. It is respectfully submitted that no new matter has been added by way of the amendments. All claims that will be pending and under consideration in the present application upon entry of the amendments are shown below. This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, a method for allocating the access of threads to a user interface context, the method comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context is configured to receive input from a user, to provide output to the user, and to maintain the context settings and the context dictionary;

determining whether the user interface context is presently being accessed by a second thread[[,]]; ~~wherein each of the first thread and the second thread includes a corresponding thread history and thread settings~~[[,]] ~~and~~

~~if~~ when the user interface context is presently being accessed by a second thread, denying the request to access the user interface context received from the first thread; and

~~if~~ when the user interface context is not presently being accessed by a second thread, performing a processes to allow for backward compatibility comprising:

(a) allowing the request to access the user interface context received from the first thread;

(b) updating a context record maintained by the first thread to reflect that access is allowed to the user interface context;

(c) verifying that the first thread has obtained exclusive access to the user interface context by checking the context record; and

~~maintaining the thread settings associated with the first thread and the second thread;~~

~~maintaining context settings in the user interface context; and~~

(d) ~~applying~~ temporarily assigning to the first thread the context settings and the context dictionary ~~of~~ maintained by the user interface context while the first thread is operating within the user interface context, wherein assigning comprises placing the context settings and the context dictionary within ~~in place of the~~ thread settings of ~~any~~ the first thread upon accessing the user interface context, and wherein ~~such that~~ settings of the context settings and dictionary information of the context dictionary are specified at a context level, rather than on a thread level.

2. (Currently Amended) The method for allocating the access of threads to user interface context of claim 1, ~~the method further~~ wherein verifying that the first thread has obtained exclusive access to the user interface context comprises ~~comprising~~:

maintaining ~~a~~ the context record associated with ~~each~~ the first thread that identifies the contexts accessed by the first thread, the most recent entry in the context record indicating the context presently being accessed by the first thread;

~~when a~~ incident to the first thread ~~accesses~~ accessing an object in the user interface context, checking the most recent entry in the context record associated with the first thread;

determining whether the most recent entry in the context record matches the user interface context ~~of~~ associated with the object being accessed; and

~~if~~ when the most recent entry in the context record does not match the user interface context ~~of~~ associated with the object being accessed, raising an exception.

3. (Canceled).

4. (Previously Presented) The method for allocating the access of threads to a user interface context of claim 13, the method further comprising restoring the thread settings when a thread departs the user interface context.

5-12. (Canceled).

13. (Currently Amended) One or more computer-storage media having computer-executable instructions embodied thereon that, when executed, perform a method for allocating the access of threads to a user interface context in a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, the method for allocating the access of threads to a user interface

context comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context comprises one or more objects, wherein the user interface context is configured to receive input from a user, to provide output to the user, and to maintain the context settings and the context dictionary, wherein the context dictionary includes information from a plurality of sources;

determining whether the user interface context is presently being accessed by a second thread[[,]]; ~~wherein each of the first thread and the second thread includes a corresponding thread history and thread settings~~[[,]] ~~and~~

~~if~~ when the user interface context is presently being accessed by a second thread, denying the request to access the user interface context received from the first thread; and

~~if~~ when the user interface context is not presently being accessed by a second thread, performing a processes to allow for backward compatibility comprising:

(a) allowing the request to access the user interface context received from the first thread;

~~maintaining the thread settings associated with the first thread and the second thread;~~

~~maintaining context settings in the user interface context;~~

(b) updating a context record maintained by the first thread to reflect that access is allowed to the user interface context;

(c) verifying that the first thread has obtained exclusive access to the user interface context by checking the context record; and

~~maintaining context dictionary in the user interface context, wherein the context dictionary comprises information from one or more sources; and~~

(d) ~~applying~~ temporarily assigning to the first thread the context settings and the context dictionary ~~of~~ maintained by the user interface context while the first thread is operating within the user interface context, wherein assigning comprises placing the context settings and the context dictionary within ~~in place of the~~ thread settings of ~~any~~ the first thread upon accessing the user interface context, and wherein ~~such that~~ settings of the context settings and dictionary information of the context dictionary are specified at a context level, rather than on a thread level.

14. (Currently Amended) The one or more computer-storage media of claim 13, the method of verifying that the first thread has obtained exclusive access to the user interface context ~~for allocating the access of threads to a user interface further~~ comprising:

maintaining ~~a~~ the context record associated with ~~each~~ the first thread that identifies the contexts accessed by the first thread, the most recent entry in the context record indicating the context presently being accessed by the first thread;

~~when a~~ incident to the first thread ~~accesses~~ accessing an object in the user interface context, checking the most recent entry in the context record associated with the first thread;

determining whether the most recent entry in the context record matches the user interface context ~~of~~ associated with the object being accessed; and

~~if~~ when the most recent entry in the context record does not match the user interface context ~~of~~ associated with the object being accessed, raising an exception.

15. (Canceled).

16. (Previously Presented) The one or more computer-storage media of claim 13, the method for allocating the access of threads to a user interface further comprising restoring the thread settings when a thread departs the user interface context.

17-24. (Canceled).